

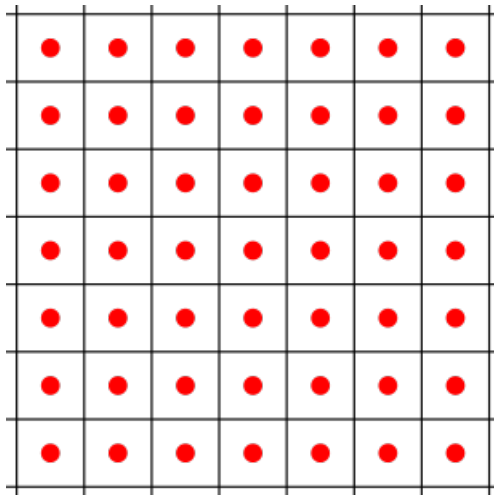
# Pokročilé spracovanie obrazu - Transformácie Obrazu

Ing. Viktor Kocur  
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

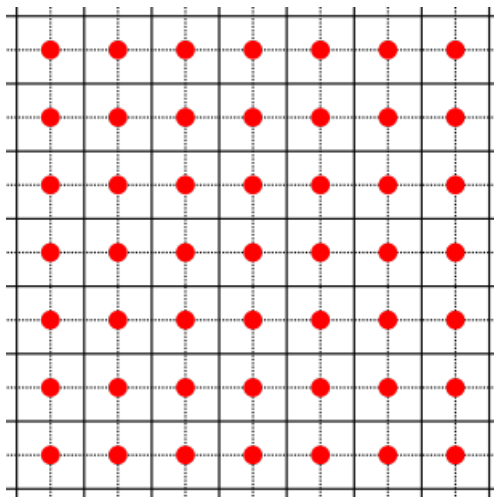
4.12.2019

# Informácia v obraze



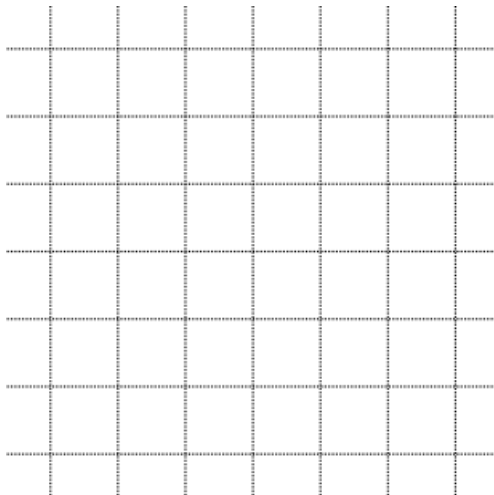
Uvažujeme, že informácia o intenzite je v strede pixela.

# Informácia v obraze

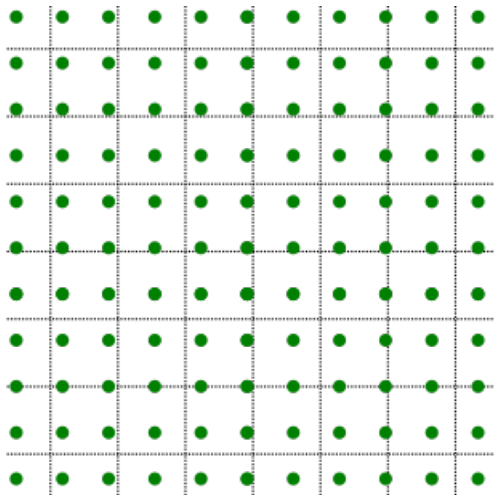


Prerušovaná mriežka nám teda nerozdejuje hranicu medzi pixelmi, ale určuje stredy pixelov.

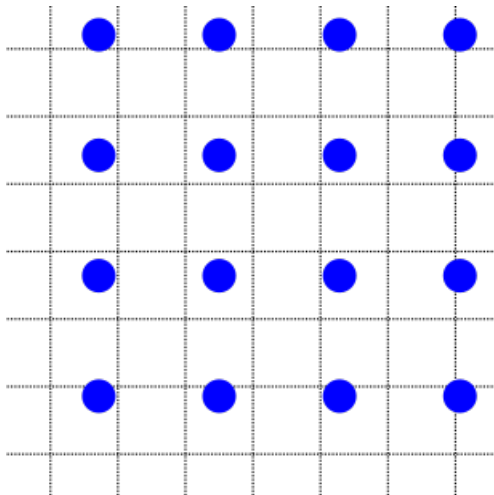
# Informácia v obraze



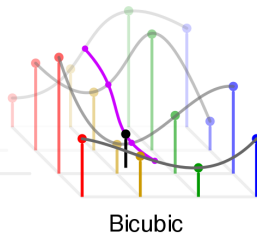
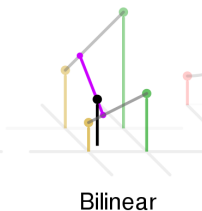
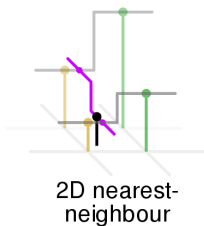
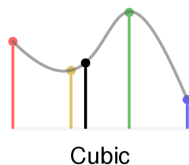
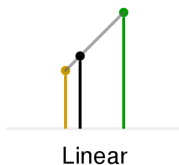
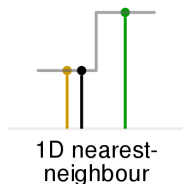
# Zväčšenie obrazu



# Zmenšenie obrazu



# Interpolácia



To aké hodnoty budú mať pixely po operácii rátame pomocou interpolácie z pixelov v okolí umiestnenia nového bodu.

# Zmena veľkosti v Matlabe

## imresize

`imresize(I, scale)` - vráti obraz po zväčšení škálovacím faktorom `scale`

## imresize

`imresize(I, [r, c])` - vráti obraz po zväčšení na rozmer  $r \times c$

## imresize

`imresize(I, s, 'method')` - vráti obraz po zväčšení, ale s použitím metódy z `'nearest'`, `'bilinear'`, `'bicubic'`.

## Úloha

Otestujte si `imresize` s rôznymi metódami pre zväčšenie `shell.png` a zmenšenie zátišia.



# Afinná transformácia

## Ako sa počíta

Transformáciu počítame pomocou zápisu kde  $\vec{y}$  predstavuje novú polohu daného pixela.

$$\vec{y} = \mathbb{A}\vec{x} + \vec{t}$$

## Využitie v obraze

V obraze nepočítame nové polohy  $\vec{y}$  na základe existujúcich polôh stredov pixelov  $\vec{x}$ , ale najprv si určíme nejakú rovnomernú množinu pre body  $\vec{y}$  a potom inverzne spočítame k nim body  $\vec{x} = \mathbb{A}^{-1}(\vec{y} - \vec{t})$ . To nám umožní jednoducho opäť použiť interpoláciu pre body v obraze.

# Príklady

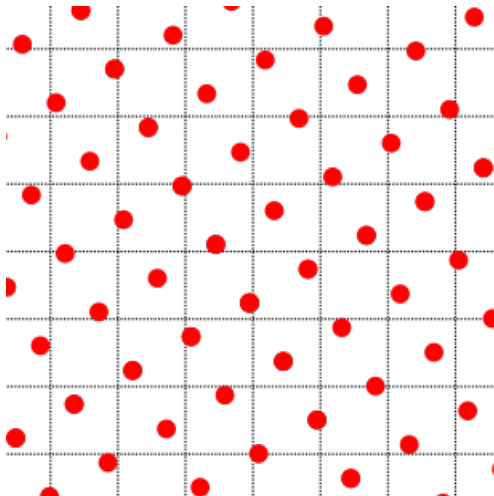
## Rotácia

$$\mathbb{A} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

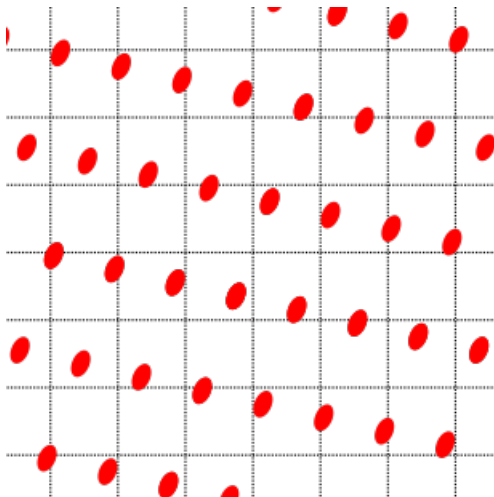
## Natiahnutie po x-ovej osi

$$\mathbb{A} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

# Rotácia



# Afinná transformácia



# Afinná transformácia v Matlabe

## imtransform

`imtransform(I, tform, interp)` - transformuje obraz `I` podľa transformačného objektu `tform` pomocou interpolačnej metódy `interp`: z 'nearest', 'bilinear', 'bicubic'.

## maketform

`maketform('affine', B)` - vráti transformačný objekt pre afinnú transformáciu. Afinná transformácia je definovaná maticou `B`, ktorá sa skladá s našej matice `A` s pridaným riadkom s vektorom  $\vec{t}$ .

## imrotate

`imrotate(I, angle)` - vráti ototovaný obrázok `I` o uhol `angle`.

# Úlohy

## Úloha

Urobte rotáciu obrázku pomocou `imrotate`. Skúste urobiť rotáciu aj pomocou afinnej transformácie.

## Úloha

Vytvorte afinnú transformáciu ktorá prehodí obraz len po x-ovej ose, alebo len po y-vej ose.

## Úloha

Otestujte si rôzne matice pre afinnú transformáciu.

# Perspektívna transformácia

## imtransform

`imtransform(I, tform, interp)` - transformuje obraz `I` podľa transformačného objektu `tform` pomocou interpolačnej metódy `interp`: z 'nearest', 'bilinear', 'bicubic'.

## maketform

`maketform('projective', U, X)` - vráti transformačný objekt pre perspektívnu transformáciu. Matice `U` a `X` majú tvar  $4 \times 2$ . Každý riadok matice `U` je transformovaný na korešpondujúci riadok matice `X`.

## Matice `U` a `X`

Maticu `U` môžeme vyrobiť pomocou volania `U = ginput(4)`. Rovnako môžeme vyrobiť aj maticu `X`, alebo ak chceme napr. obraz rektifikovať (zarovnať s osami) tak si vyrobíme maticu ktorá na každom riadku bude mať pozíciu rohu obdĺžnika.

# Úlohy

## Úloha

V obrázku qr.jpg použite perspektívnu transformáciu tak, aby boli QR kódy zarovnané s osami. Podobne zarovnajte aj obrázok book.jpg.

## Úloha

V obrázku road.png použite perspektívnu transformáciu tak, aby boli vodiace čiary zarovnané s y-ovou osou. Je toto zadanie jednoznačné z hľadiska výsledného obrazu?